(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 503 684 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
01.07.1998 Bulletin 1998/27

(51) Int Cl.[6]: **G10L 9/14**

(21) Application number: **92108904.1**

(22) Date of filing: **06.04.1988**

(54) **Adaptive filtering method for speech and audio**

Verfahren zur adaptiven Filterung von Sprach- und Audiosignalen

Procédé de filtrage adaptatif de la parole et de signaux audio

(84) Designated Contracting States:
DE FR GB IT

(30) Priority: **06.04.1987 US 35615**

(43) Date of publication of application:
16.09.1992 Bulletin 1992/38

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
88303038.9 / 0 294 020

(73) Proprietor: **VOICECRAFT, INC.**
Goleta, CA 93117 (US)

(72) Inventors:
- **Chen, Juin-Hwey**
  Canton, MA 02021 (US)
- **Gersho, Allen**
  Goleta, CA 93117 (US)

(74) Representative: **Tomlinson, Kerry John**
Frank B. Dehn & Co.,
European Patent Attorneys,
179 Queen Victoria Street
London EC4V 4EL (GB)

(56) References cited:
GB-A- 2 188 820

- **ICASSP'86 (IEEE-IECEJ-ASJ INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Tokyo, 7th - 11th April 1986), vol. 4, pages 3071-3074, IEEE, New York, US; Y. YATSUZUKA et al.: "A variable rate coding by APC with maximum likelihood quantization from 4.8 kbit/s to 16 kbit/s"**
- **ICASSP'87 (INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, Dallas, Texas, 6th - 9th April 1987), vol. 4, pages 2185-2188,IEEE, New York; US; J.-H. CHEN et al.: "Real-time vector APC speech coding at 4800 BPS with adaptive postfiltering"**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 503 684 B1

Description

This invention relates an adaptive filtering method for enhancing digitally processed speech or audio signals e.g. from a real-time coder for compression of digitally encoded speech or audio signals for transmission or storage, or more particularly from a real-time vector adaptive predictive coding system.

In the past few years, most research in speech coding has focused on bit rates from 16 kb/s down to 150 bits/s. At the high end of this range, it is generally accepted that toll quality can be achieved at 16 kb/s by sophisticated waveform coders which are based on scalar quantization. N.S. Jayant and P. Noll, Digital Coding of Waveforms, Prentice-Hall Inc., Englewood Cliffs, N.J., 1984. At the other end, coders (such as linear-predictive coders) operating at 2400 bits/s or below only give synthetic-quality speech. For bit rates between these two extremes, particularly between 4.8 kb/s and 9.6 kb/s, neither type of coder can achieve high-quality speech. Part of the reason is that scalar quantization tends to break down at a bit rate of 1 bit/sample. Vector quantization (VQ), through its theoretical optimality and its capability of operating at a fraction of one bit per sample, offers the potential of achieving high-quality speech at 9.6 kb/s or even at 4.8 kb/s. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," Proc. IEEE, Vol. 73, No. 11, November 1985.

Vector quantization (VQ) can achieve a performance arbitrarily close to the ultimate rate-distortion bound if the vector dimension is large enough. T. Berger, Rate Distortion Theory, Prentice-Hall Inc., Englewood Cliffs,N.J., 1971. However, only small vector dimensions can be used in practical systems due to complexity considerations, and unfortunately, direct waveform VQ using small dimensions does not give adequate performance. One possible way to improve the performance is to combine VQ with other data compression techniques which have been used successfully in scalar coding schemes.

In speech coding below 16 kb/s, one of the most successful scalar coding schemes is Adaptive Predictive Coding (APC) developed by Atal and Schroeder [B.S. Atal and M.R. Schroeder, "Adaptive Predictive Coding of Speech Signals, " Bell Syst. Tech. J., Vol. 49, pp. 1973-1986, October 1970; B.S. Atal and M.R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," IEEE Trans. Acoust., Speech, Signal Proc., Vol. ASSP-27, No. 3, June 1979; and B.S. Atal, "Predictive Coding of Speech at Low Bit Rates," IEEE Trans. Comm., Vol. COM-30, No. 4, April 1982]. It is the combined power of VQ and APC that led to the development of the present invention, a Vector Adaptive Predictive Coder (VAPC). Such a combination of VQ and APC will provide high-quality speech at bit rates between 4.8 and 9.6 kb/s, thus bridging the gap between scalar coders and VQ coders.

The basic idea of APC is to first remove the redundancy in speech waveforms using adaptive linear predictors, and then quantize the prediction residual using a scalar quantizer. In VAPC, the scalar quantizer in APC is replaced by a vector quantizer VQ. The motivation for using VQ is two-fold. First, although linear dependency between adjacent speech samples is essentially removed by linear prediction, adjacent prediction residual samples may still have nonlinear dependency which can be exploited by VQ. Secondly, VQ can operate at rates below one bit per sample. This is not achievable by scalar quantization, but it is essential for speech coding at low bit rates.

The vector adaptive predictive coder (VAPC) has evolved from APC and the vector predictive coder introduced by V. Cuperman and A. Gersho, "Vector Predictive Coding of Speech at 16 kb/s," IEEE Trans. Comm., Vol. COM-33, pp. 685-696, July 1985. VAPC contains some features that are somewhat similar to the Code-Excited Linear Prediction (CELP) coder by M.R. Schroeder, B.S. Atal, "Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates," Proc. Int'l. Conf. Acoustics, Speech, Signal Proc., Tampa, March 1985, but with much less computational complexity.

In computer simulations, VAPC gives very good speech quality at 9.6 kb/s, achieving 18 dB of signal-to-noise ratio (SNR) and 16 dB of segmental SNR. At 4.8- kb/s, VAPC also achieves reasonably good speech quality, and the SNR and segmental SNR are about 13 dB and 11.5 dB, respectively. The computations required to achieve these results are only in the order of 2 to 4 million flops per second (one flop, a floating point operation, is defined as one multiplication, one addition, plus the associated indexing), well within the capability of today's advanced digital signal processor chips. VAPC may become a low-complexity alternative to CELP, which is known to have achieved excellent speech quality at an expected bit rate around 4.8 kb/s but is not presently capable of being implemented in real-time due to its astronomical complexity. It requires over 400 million flops per second to implement the coder. In terms of the CPU time of a supercomputer CRAY-1, CELP requires 125 seconds of CPU time to encode one second of speech. There is currently a great need for a real-time, high-quality speech coder operating at encoding rates ranging from 4.8 to 9.6 kb/s. In this range of encoding rates, the two coders mentioned above (APC and CELP) are either unable to achieve high quality or too complex to implement. In contrast, the system of EP-A-0294020, from which the present application is divided, combines Vector Quantization (VQ) with the advantages of both APC and CELP, is able to achieve high-quality speech with sufficiently low complexity for real-time coding.

The noise-masking effect of human auditory perception is exploited in many speech coders by using noise spectral shaping. However, in noise spectral shaping, lowering noise components at certain frequencies can only be achieved at the price of increased noise components at other frequencies. Therefore, at bit-rates where the average noise level

is quite high, it is very difficult, if not impossible, to force noise below the masking threshold at all frequencies. Since speech formants are much more important to perception than spectral valleys, the goal is to preserve the formant information by keeping the noise in the formant regions as low as is practical during encoding. Of course, in this case, the noise components in spectral valleys may exceed the threshold; however, these noise components can be attenuated later by a postfilter. In performing such postfiltering, the speech components in spectral valleys will also be attenuated. Fortunately, the limen, or just noticeable difference for the intensity of spectral valleys can be quite large. Therefore, by attenuating the components in spectral valleys, the postfilter only introduces minimal distortion in the speech signal, but it achieves a substantial noise reduction.

Adaptive postfiltering has been used successfully in enhancing ADCM-coded speech. Such a postfilter reduces the overall noise level; however, sufficient noise reduction can only be achieved with severe muffling in the filtered speech. This is due to the fact that the frequency response of this postfilter generally has a lowpass spectral tilt for voiced speech.

At a conference in Tokyo sponsored by the IEEE Acoustics, Speech and Signal Processing Society, the Institute of Electronics and Communications Engineers of Japan and the Acoustical Society of Japan, a variable rate APC coding system with Maximum Likelihood Quantization (MLQ) was presented [Y. Yatsusuka et al., "A Variable Rate Coding by APC with Maximum Likelihood Quantization From 4.8 KBit/s to 16 KBit/s, "Proc. ICASSP '86. Vol. 4, pp. 3071-74, April 6, 1986] with adaptive noise-shaping filters in both the encoder and the decoder. An all-pole filter is inserted around an adaptive quantizer in the coder and another in the decoder after decoding both filters consisting of long- and short-term predictors.

An object of this inventionis to provide adaptive postfiltering of a speech or audio signal that has been corrupted by noise resulting from a coding system or other sources of degradation so as to enhance the perceived quality of said speech or audio signal.

According to the invention there is provided an adaptive filtering method for enhancing digitally processed speech or audio signals at a receiver by filtering said digitally processed signals with short-delay filtering, said short-delay filtering being controlled by predetermined linear-predictive coefficient (LPC) parameters; characterised in that said short-delay filtering uses a pole-zero transfer function consisting of the ratio of two all-pole transfer functions, with the zeros of said pole-zero transfer function having smaller radii than corresponding poles.

The preferred embodiment provides postfiltering for use with a system which approximates each vector of K speech samples by using each of M fixed vectors stored in a VQ codebook to excite a time-varying synthesis filter and picking the best synthesized vector that minimizes a perceptually meaningful distortion measure. The original sampled speech is first buffered and partitioned into vectors and frames of vectors, where each frame is partitioned into N vectors, each vector having K speech samples. Predictive analysis of pitch-filtering parameters (P) linear-predictive coefficient filtering parameters (LPC), perceptual weighting filter parameters (W) and residual gain scaling factor (G) for each of successive frames of speech is then performed. The parameters determined in the analyses are quantized and reset every frame for processing each input vector s_n in the frame, except the perceptual weighting parameter. A perceptual weighting filter responsive to the parameters W is used to help select the VQ vector that minimizes the perceptual distortion between the coded speech and the original speech. Although not quantized, the perceptual weighting filter parameters are also reset every frame.

After each frame is buffered and the above analysis is completed at the beginning of each frame, M zero-state response vectors are computed and stored in a zero-state response codebook. These M zero-state response vectors are obtained by setting to zero the memory of an LPC synthesis filter and a perceptual weighting filter in cascade after a scaling unit controlled by the factor G, and controlling the respective filters with the quantized LPC filter parameters and the unquantized perceptual weighting filter parameters, and exciting the cascaded filters using one predetermined and fixed codebook vector at a time. The output vector of the cascaded filters for each VQ codebook vector is then stored in the corresponding address, i.e., is assigned the same index of a temporary zero-state response codebook as of the VQ codebook. In encoding each input speech vector s_n within a frame, a pitch prediction $\hat{s}_n$ of the vector is determined by processing the last vector encoded as an index code through a scaling unit, LPC synthesis filter and pitch predictor filter controlled by the parameters QG, QLPC, QP and QPP for the frame. In addition, the zero-input response of the cascaded filters (the ringing from excitation of a previous vector) is first set in a filter. Once the pitch-predicted vector $\hat{s}_n$ is subtracted from the input signal vector s_n, and a difference vector d_n is passed through the perceptual weighting filter to produce a filtered difference vector f_n, the zero-input response vector in the aforesaid filter is subtracted from the perceptual weight filtered difference vector f_n, and the resulting vector v_n is compared with each of the M stored zero-state response vectors in search of the one having a minimum difference Δ or distortion.

The index (address) of the zero-state response vector that produces the smallest distortion, i.e., that is closest to v_n, identifies the best vector in the permanent codebook. Its index (address) is transmitted as the compressed code for the vector, and used by a receiver which has an identical VQ codebook as the transmitter to find the best-match vector. In the transmitter, that best-match vector is used at the time of transmission of its index to excite the LPC synthesis filter and pitch prediction filter to generate an estimate $\hat{s}_n$ of the next speech vector. The best-match vector

is also used to excite the zero-input response filter to set it for the next speech vector s_n as described above. The indices of the best-match vector for a frame of vectors are combined in a multiplexer with the frame analysis information hereinafter referred to as "side information," comprised of the indices of parameters which control pitch, pitch predictor and LPC predictor filtering and the gain used in the coding process, in order that it may be used by the receiver in decoding the vector indices of a frame into vectors using a codebook identical to the permanent codebook at the transmitter. This side information is preferably transmitted through the multiplexer first, once for each frame of VQ indices that follow, but it would be possible to first transmit a frame of vector indices, and then transmit the side information since the frames of vector indices will require some buffering in either case; the difference is only in some initial delay at the beginning of speech or audio frames transmitted in succession. The resulting stream of multiplexed indices are transmitted over a communication channel to a decoder, or stored for later decoding.

In the decoder, the bit stream is first demultiplexed to separate the side information from the indices that follow. Each index is used at the receiver to extract the corresponding vector from the duplicate codebook. The extracted vector is first scaled by the gain parameter, using a table to convert the gain index to the appropriate scaling factor, and then used to excite cascaded LPC synthesis and pitch synthesis filters controlled by the same side information used in selecting the best-match index utilizing the zero-state response codebook in the transmitter. The output of the pitch synthesis filter is the coded speech, which is perceptually close to the original speech. All of the side information, except the gain information, is used in an adaptive postfilter to enhance the quality of the speech synthesized. This postfiltering technique may be used to enhance any voice or audio signal. All that would be required is an analysis section to produce the parameters used to make the postfilter adaptive.

Although reference is made hereinafter only to speech, the invention described and claimed is applicable to audio waveforms or to sub-band filtered speech or audio waveforms.

The present application is a divisional application from European Patent Application 88303038.9 filed on 6 April 1988.

An example of the invention will now be described with reference to the accompanying drawings.

FIG. 1a is a block diagram of a Vector Adaptive Predictive Coding (VAPC) processor, and FIG. 1b is a block diagram of a receiver for the encoded speech transmitted by the system of **FIG. 1a.**

FIG. 2 is a schematic diagram that illustrates the adaptive computation of vectors for a zero-state response codebook in the system of **FIG. 1a.**

FIG. 3 is a block diagram of an analysis processor in the system of **FIG. 1a.**

FIG. 4 is a block diagram of an adaptive post filter according to the present invention, which may be used in the receiver of **FIG. 1b**.

FIG. 5 illustrates the LPC spectrum and the corresponding frequency response of an all-pole postfilter $1/[\hat{1}\text{-}P(z/\alpha)]$ for different values of α. The offset between adjacent plots is 20 dB.

FIG. 6 illustrates the frequency responses of the postfilter $[1\text{-}\mu z^{-1}][1\text{-}\hat{P}(z/\beta)]/[1\text{-}\hat{P}(z/\alpha)]$ corresponding to the LPC spectrum shown in **FIG. 5.** In both plots, α=0.8 and β=0.5. The offset between the two plots is 20 dB.

Referring to **FIG. 1a,** original speech samples, s_n in digital form from sampling analog-to-digital converter 10 are received by an analysis processor **11** which partitions them into vectors s_n of K samples per vector, and into frames of N vectors per frame. The analysis processor stores the samples in a dual buffer memory which has the capacity for storing more than one frame of vectors, for example two frames of 8 vectors per frame, each vector consisting of 20 samples, so that the analysis processor may compute parameters used for coding the following frame. As each frame is being processed out of one buffer, a new frame coming in is stored in the other buffer so that when processing of a frame has been completed, there is a new frame buffered and ready to be processed.

The analysis processor determines the parameters of filters employed in the Vector Adaptive Predictive Coding technique.

These parameters are transmitted through a multiplexer **12** as side information just ahead of the frame of vector codes generated with the use of a vector quantized (VQ) permanent codebook **13** and a zero-state response (ZSR) codebook **14.** The side information conditions the receiver to properly filter decoded vectors of the frame. The analysis processor **11** also computes other parameters used in the encoding process. The latter are represented in **FIG. 1a** by dashed lines, and consist of sets of parameters which are designated W for a perceptual weighting filter **18,** a quantized LPC predictor QLPC for an LPC synthesis filter **15,** and quantized pitch QP and pitch predictor QPP for a pitch synthesis filter **16.** Also computed by the analysis processor is a scaling factor G for control of a scaling unit **17.** The four quantized parameters transmitted as side information are encoded using a quantizing table as the quantized pitch index, pitch predictor index, LPC predictor index and gain index. The manner in which the analysis processor computes all of these parameters will be described with reference to **FIG. 3.**

The multiplexer **12** preferably transmits the side information as soon as it is available, although it could follow the frame of encoded input vectors, and while that is being done, **M** zero-state response vectors are computed for the zero-state response (ZSR) codebook **14** in a manner illustrated in **FIG. 2,** which is to process each vector in the VQ codebook, **13** e.g., 128 vectors, through a gain scaling unit **17',** an LPC synthesis filter **15',** and perceptual weighting

filters **18'** corresponding to the gain scaling unit **17**, the LPC synthesis filter 15, and perceptual weighting filter **18** in the transmitter (**FIG. 1a**). Ganged commutating switches S_1 and S_2 are shown to signify that each fixed VQ vector processed is stored in memory locations of the same index (address) in the ZSR codebook.

At the beginning of each vector processing, the initial conditions of the cascaded filters **15'** and **18'** are set to zero. This simulates what the cascaded filters **15'** and **18'** will do with no previous vector present from its corresponding VQ codebook. Thus, if the output of a -zero-input response filter **19** in the transmitter (**FIG. 1a**) is held or stored,

at each step of computing the VQ code index (to transmit for each vector of a frame), it is possible to simplify encoding the speech vectors by subtracting the zero-state response output from the vector f_n. In other words, assuming M=128, there are 128 different vectors permanently stored in the VQ codebook to use in coding the original speech vectors s_n. Then every one of the 128 VQ vectors is read out in sequence, fed through the scaling unit **17'**, the LPC synthesis filter **15'**, and the perceptual weighting filter **18'** without any history of previous vector inputs by resetting those filters at each step. The resulting filter output vector is then stored in a corresponding location in the zero-state response codebook. Later, while encoding input signal vectors s_n by finding the best match between a vector v_n and all of the zero state response vector codes, it is necessary to subtract from a vector f_n derived from the perceptual weighting filter a value that corresponds to the effect of the previously selected VQ vector. That is done through the zero-input response filter **19.** The index (address) of the best match is used as the compressed vector code transmitted for the vector s_n. Of the 128 zero-state response vectors, there will be only one that prothe best match, i.e., least distortion. Assume it is in location 38 of the zero-state response codebook as determined by a computer **20** labeled "compute norm." An address register **20a** will store the index **38.** It is that index that is then transmitted as a VQ index to the receiver shown in **FIG. 1b.**

In the receiver, a demultiplexer **21** separates the side information which, conditions the receiver with the same parameters as corresponding filters and scaling unit of the transmitter. The receiver uses a decoder **22** to translate the parameters indices to parameter values. The VQ index for each successive vector in the frame addresses a VQ codebook **23** which is identical to the fixed VQ codebook **13** of the transmitter. The LPC synthesis filter **24,** pitch synthesis filter **25,** and scaling unit **26** are conditioned by the same parameters which were used in computing the zero-state codebook values, and which were in turn used in the process of selecting the encoding index for each input vector. At each step of finding and transmitting an encoding index, the zero-input response filter 19 computes from the VQ vector at the location of the index transmitted a value to be subtracted from the input vector f_n to present a zero-input response to be used in the best-match search.

There are various procedures that may be used to determine the best match for an input vector s_n. The simplest is to store the resulting distortion between each zero-state response vectorcode output and the vector v_n with the index of that. Assuming there are 128 vectorcodes stored in the codebook **14,** there would then be 128 resulting distortions stored in a best address computer **20.** Then, after all have been stored, a search is made in the computer **20** for the lowest distortion value. Its index is then transmitted to the receiver as an encoded vector via the multiplexer **12,** and to the VQ codebook for reading the corresponding VQ vector to be used in the processing of the next input vector s_n.

In summary, it should be noted that the VQ codebook is used (accessed) in two different steps: first, to compute vector codes for the zero-state response codebook at the beginning of each frame, using the LPC synthesis and perceptual weighting filter parameters determined for the frame; and second, to excite the filters **15** and **16** through the scaling unit **17** while searching for the index of the best-match vector, during which the estimate $\hat{s}_n$ thus produced is subtracted from the input vector s_n. The difference d_n is used in the best-match search.

As the best match for each input vector s_n is found, the corresponding predetermined and fixed vector from the VQ codebook is used to reset the zero input response filter **19** for the next vector of the frame. The function of the zero-input response filter **19** is thus to find the residual response of the gain scaling unit **17'** and filters **15'** and **18'** to previously selected vectors from the VQ codebook. Thus, the selected vector is not transmitted; only its index is transmitted. At the receiver its index is used to read out the selected vector from a VQ codebook **23** identical to the VQ codebook **13** in the transmitter.

The zero-input response filter **19** is the same filtering operation that is used to generate the ZSR codebook, namely the combination of a gain G, an LPC synthesis filter and a weighting filter, as shown in **FIG. 2.** Once a best codebook vector match is determined, mined, the best-match vector is applied as an input to this filter (sample by sample, sequentially). An input switch s_i is closed and an output switch s_o is open during this time so that the first K output samples are ignored. (K is the dimension of the vector and a typical value is 20.) As soon as all K samples have been applied as input to the filter, the filter input switch s_i is opened and the output switch s_o is closed. The next K samples of the vector f_n, the output of the perceptual weighting filter, begin to arrive and are subtracted from the samples of the vector f_n. The difference so generated is a set of K samples forming the vector v_n which is stored in a static register for use in the ZSR codebook search procedure. In the ZSR codebook search procedure, the vector v_n is subtracted from each vector stored in the ZSR codebook, and the difference vector Δ is fed to the computer **20** together with the index (or stored in the same order), thereby to imply the index of the vector out of the ZSR codebook. The computer **20** then determines which difference is the smallest, i.e., which is the best match between the vector v_n and each vector stored

temporarily (for one frame of input vectors s_n). The index of that best-match vector is stored in a register **20a.** That index is transmitted as a vectorcode and used to address the VQ codebook to read the vector stored there into the scaling unit **17**, as noted above. This search process is repeated for each vector in the ZSR code-book, each time using the same vector v_n. Then the best vector is determined.

Referring now to **FIG. 1b**, it should be noted that the output of the VQ codebook **23**, which precisely duplicates the VQ codebook **13** of the transmitter, is identical to the vector extracted from the best-match index applied as an address to the VQ codebook **13**; the gain unit **26** is identical to the gain unit **17** in the transmitter, and filters **24** and **25** exactly duplicate the filters **15** and **16,** respectively, except that at the receiver, the approximation $\bar{s}_n$ rather than the prediction $\hat{s}_n$ is taken as the output of the pitch synthesis filter **25.** The result, after converting from digital to analog form, is synthesized speech that reproduces the original speech with very good quality.

It has been found that by applying an adaptive postfilter **30** to the synthesized speech before converting it from digital to analog form, the perceived coding noise may be greatly reduced without introducing significant distortion in the filtered speech. **FIG. 4** illustrates the preferred organization of the adaptive postfilter as a long-delay filter **31** and a short-delay filter **32.** Both filters are adaptive in that the parameters used in them are those received as side information from the transmitter, except for the gain parameter, G. The basic idea of adaptive postfiltering is to attenuate the frequency components of the coded speech in spectral valley regions. At low bit rates, a considerable amount of perceived coding noise comes from spectral valley regions where there are no strong resonances to mask the noise. The postfilter attenuates the noise components in spectral valley regions to make the coding noise less perceivable. However, such filtering operation inevitably introduces some distortion to the shape of the speech spectrum. Fortunately, our ears are not very sensitive to distortion in spectral valley regions; therefore, adaptive postfiltering only introduces very slight distortion in perceived speech, but it significantly reduces the perceived noise level. The adaptive postfilter will be described in greater detail after first describing in more detail the analysis of a frame of vectors to determine the side information.

Referring now to **FIG. 3**, it shows the organization of the initial analysis of block **11** in **FIG. 1a.** The input speech samples s_n are first stored in a buffer **40** capable of storing, for example, more than one frame of 8 vectors, each vector having 20 samples.

Once a frame of input vectors s_n has been stored, the parameters to be used, and their indices to be transmitted as side information, are determined from that frame and at least a part of the previous frame in order to perform analysis with information from more than the frame of interest. The analysis is carried out as shown using a pitch detector **41,** pitch quantizer **42** and a pitch predictor coefficient quantizer **43.** What is referred to as "p-itch" applies to any observed periodicity in the input signal, which may not necessarily correspond to the classical use of "pitch" corresponding to vibrations in the human vocal folds. The direct output of the speech is also used in the pitch predictor coefficient quantizer **43.** The quantized pitch (QP) and quantized pitch predictor (QPP) are used to compute a pitch-prediction residual in block **44,** and as control parameters for the pitch synthesis filter **16**-used as a predictor in **FIG. 1a**. Only a pitch index and a pitch prediction index are included in the side information to minimize the number of bits transmitted. At the receiver, the decoder 22 will use each index to prothe corresponding control parameters for the pitch synthesis filter **25**.

The pitch-prediction residual is stored in a buffer **45** for LPC analysis in block **46.** The LPC predictor from the LPC analysis is quantized in block **47.** The index of the quantized LPC predictor is transmitted as a third one of four pieces of side information, while the quantized LPC predictor is used as a parameter for control of the LPC synthesis filter **15,** and in block **48** to compute the rms value of the LPC predictive residual. This value (unquantized residual gain) is then quantized in block **49** to provide gain control G in the scaling unit **17** of **FIG. 1a.** The index of the quantized residual gain is the fourth part of the side information transmitted.

In addition to the foregoing, the analysis section provides LPC analysis in block **50** to produce an LPC predictor from which the set of parameters W for the perceptual weighting filter **18 (FIG. 1a)** is computed in block **51.**

The adaptive postfilter **30** in **FIG. 1b** will now be described with reference to **FIG. 4.** It consists of a long-delay filter **31** and a short-delay filter 32 in cascade. The long-delay filter is derived from the decoded pitch-predictor information available at the receiver. It attenuates frequency components between pitch harmonic frequencies. The short-delay filter is derived from LPC predictor information, and it attenuates the frequency components between formant frequencies.

The noise masking effect of human auditory perception, recognized by M.R. Schroeder, B.S. Atal, and J.L. Hall, "Optimizing Digital Speech Coders by Exploiting Masking Properties of the Human Ear," J. Acoust. Soc. Am., Vol. 66, No. 6, pp. 1647-1652, December 1979, is exploited in VAPC by using noise spectral shaping. However, in noise spectral shaping, lowering noise components at certain frequencies can only be achieved at the price of increased noise components at other frequencies. [B.S. Atal and M.R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," IEEE Trans. Acoust., Speech, and Signal Processing, Vol. ASSP-27, No. 3, pp. 247-254, June 1979] Therefore, at bit rates as low as 4800 bps, where the average noise level is quite high, it is very difficult, if not impossible, to force noise below the masking threshold at all frequencies. Since speech formants are much more important to

perception than spectral valleys, the approach of the present invention is to preserve the formant information by keeping the noise in the formant regions as low as is practical during encoding. Of course, in this case, the noise components in spectral valleys may exceed the threshold; however, these noise components can be attenuated later by the postfilter **30**. In performing such postfiltering, the speech components in spectral valleys will also be attenuated. Fortunately, the limen, or "just noticeable difference," for the intensity of spectral valleys can be quite large [J.L. Flanagan, Speech Analysis, Synthesis, and Perception, Academic Press, New York, 1972]. Therefore, by attenuating the components in spectral valleys, the postfilter only introduces minimal distortion in the speech signal, but it achieves a substantial noise reduction.

Adaptive postfiltering has been used successfully in enhancing ADPCM-coded speech. See V. Ramamoorthy and J.S. Jayant, "Enhancement of ADPCM Speech by Adaptive Postfiltering," AT&T Bell Labs Tech. J., pp. 1465-1475, October 1984; and N.S. Jayant and V. Ramamoorthy, "Adaptive Postfiltering of 16 kb/s-ADPCM Speech," Proc. ICASSP, pp. 829-832, Tokyo, Japan, April 1986. The postfilter used by Ramamoorthy, et al., supra, is derived from the two-pole six-zero ADPCM synthesis filter by moving the poles and zeros radially toward the origin. If this idea is extended directly to an all-pole LPC synthesis filter $1/[1-\hat{P}(z)]$, the result is $1/[1-\hat{P}(z/\alpha)]$ as the corresponding postfilter, where $0<\alpha<1$. Such an all-pole postfilter indeed reduces the perceived noise level; however, sufficient noise reduction can only be achieved with severe muffling in the filtered speech. This is due to the fact that the frequency response of this all-pole postfilter generally has a lowpass spectral tilt for voiced speech.

The spectral tilt of the all-pole postfilter $1/[1-\hat{P}(z/\alpha)]$ can be easily reduced by adding zeros having the same phase angles as the poles but with smaller radii. The transfer function Qf the result ing pole-zero postfilter **32a** has the form

$$H(z) = \frac{1-\hat{P}(z/\beta)}{1-\hat{P}(z/\alpha)}, \quad 0<\beta<\alpha<1 \tag{1}$$

where α and β are coefficients empirically determined, with some tradeoff between spectral peaks being so sharp as to produce chirping and being so low as to not achieve any noise reduction. The frequency response of H(z) can be expressed as

$$20 \log|H(e^{jw})| = 20 \log \frac{1}{|1-\hat{P}(e^{jw}/\alpha)|} - 20 \log \frac{1}{|1-\hat{P}(e^{jw}/\beta)|} \tag{2}$$

Therefore, in logarithmic scale, the frequency response of the pole-zero postfilter H(z) is simply the difference between the frequency responses of two all-pole postfilters.

Typical values of α and β are 0.8 and 0.5, respectively. From **FIG. 5**, it is seen that the response for α=0.8 has both formant peaks and spectral tilt, while the response for α=0.5 has spectral tilt only. Thus, with α=0.8 and β=0.5 in Equation 2, we can at least partially remove the spectral tilt by subtracting the response for α=0.5 from the response for α=0.8. The resulting frequency response of H(z) is shown in the upper plot of **FIG. 6.**

In informal listening tests, it has been found that the muffling effect was significantly reduced after the numerator term $[1-\hat{P}(z/\beta)]$ was included in the transfer function H(z). However, the filtered speech remained slightly muffled even with the spectral-tilt compensating term $[1-\hat{P}(z/\beta)]$. To further reduce the muffling effect, a first-order filter **32b** was added which has a transfer function of $[1-\mu z^{-1}]$, where μ is typically 0.5. Such a filter provides a slightly highpassed spectral tilt and thus helps to reduce muffling. This first-order filter is used in cascade with H(z), and a combined frequency response with μ=0.5 is shown in the lower plot of **FIG. 6.**

The short-delay postfilter **32** just described basically amplifies speech formants and attenuates inter-formant valleys. To obtain the ideal postfilter frequency response, we also have to amplify the pitch harmonics and attenuate the valleys between harmonics. Such a characteristic of frequency response can be achieved with a long-delay postfilter using the information in the pitch predictor.

In VAPC, we use a three-tap pitch predictor; the pitch synthesis filter corresponding to such a pitch predictor is not guaranteed to be stable. Since the poles of such a synthesis filter may be outside the unit circle, moving the poles toward the origin may not have the same effect as in a stable LPC synthesis filter. Even if the three-tap pitch synthesis filter is stabilized, its frequency response may have an undesirable spectral tilt. Thus, it is not suitable to obtain the long-delay postfilter by scaling down the three tap weights of the pitch synthesis filter.

With both poles and zeroes, the long-delay postfilter can be chosen as

$$H_l(z) = C_g \frac{1+\gamma z^{-p}}{1-\lambda z^{-p}} \quad (3)$$

where p is determined by pitch analysis, and C_g is an adaptive scaling factor.

Knowing the information provided by a single or three-tap pitch predictor as the value b_2 or the sum of $b_1+b_2+b_3$, the factors γ and λ are determined according to the following formulas:

$$\gamma = C_z f(x),\ \lambda = C_p f(x),\ 0 < C_z, C_p < 1 \quad (4)$$

where

$$f(x) = \begin{cases} 1 \text{ if } x > 1 \\ x \text{ if } U_{th} \le x \le 1 \\ 0 \text{ if } x < U_{th} \end{cases} \quad (5)$$

where Uth is a threshold value (typically 0.6) determined empirically, and x can be either b_2 or $b_1+b_2+b_3$ depending on whether a one-tap or a three-tap pitch predictor is used. Since a quantized three-tap pitch predictor is preferred and therefore already available at the VAPC receiver, x is chosen as

$$\sum_{i=1}^{3} b_i,$$

in VAPC postfiltering. On the other hand, if the postfilter is used elsewhere to enhance noisy input speech; a separate pitch analysis is needed, and x may be chosen as a single value b_2 since a one-tap pitch predictor suffices. (The value b_2 when used alone indicates a value from a single-tap predictor, which in practice would be the same as a three-tap predictor when b_1 and b_3 are set to zero.)

The goal is to make the power of {y(n)} about the same as that of {s(n)}. An appropriate scaling factor is chosen as

$$C_g = \frac{1 - \lambda/x}{1 + \gamma/x} \quad (6)$$

The first-order filter **32b** can also be made adaptive to better track the change in the spectral tilt of H(z). However, it has been found that even a fixed filter with μ=0.5 gives quite satisfactory results. A fixed value of μ may be determined empirically.

To avoid occasional large gain excursions, an automatic gain control (AGC) was added at the output of the adaptive postfilter. The purpose of AGC is to scale the enhanced speech such that it has roughly the same power as the unfiltered noisy speech. It is comprised of a gain (volume) estimator **33** operating on the speech input s(n), a gain (volume) estimator **34** operating on the postfiltered output r(n), and a circuit **35** to compute a scaling factor as the ratios of the two gains. The postfiltering output r(n) is then multiplied by this ratio in a multiplier **36**. AGC is thus achieved by estimating the power of the unfiltered and filtered speech separately and then using the ratio of the two values as the scaling factor. Let {s(n)} be the sequence of either unfiltered or filtered speech samples; then, the speech power $\sigma^2(n)$ is estimated by using

$$\sigma^2(n)=\zeta\ \sigma^2(n-1)+(1-\zeta)s^2(n),\ 0<\zeta<1. \quad (7)$$

A suitable value of ξ is 0.99.

The complexity of the postfilter described in this section is only a small fraction of the overall complexity of the rest

of the VAPC system, or any other coding system that may be used. In simulations, this postfilter achieves significant noise reduction with almost negligible distortion in speech. To test for possible distorting effects, the adaptive postfiltering operation was applied to clean, uncoded speech and it was found that the unfiltered original and its filtered version sound essentially the same, indicating that the distortion introduced by this postfilter is negligible.

It should be noted -that although this novel postfiltering technique was developed for use with the VAPC system, its applications are not restricted to use with it. In fact, this technique can be used not only to enhance the quality of any noisy digital speech signal but also to enhance the decoded speech of other speech coders when provided with a buffer and analysis section for determining the parameters.

What has been disclosed in the parent application from which the present application is divided, is a real-time Vector Adaptive Predictive Coder (VAPC) for speech or audio which may be implemented with software using the commercially available AT&T DSP32 digital processing chip). In its newest version, this chip has a processing power of 6 million instructions per second (MIPS). To facilitate implementation for real-time speech coding, a simplified version of the 4800 bps VAPC is available. This simplified version has a much lower complexity, but gives nearly the same speech quality as a full complexity version.

Claims

1. An adaptive filtering method for enhancing digitally processed speech or audio signals at a receiver by filtering said digitally processed signals with short-delay filtering, said short-delay filtering being controlled by predetermined linear-predictive coefficient (LPC) parameters; characterised in that said short-delay filtering uses a pole-zero transfer function consisting of the ratio of two all-pole transfer functions, with the zeros of said pole-zero transfer function having smaller radii than corresponding poles.

2. An adaptive filtering method as defined in claim 1 wherein said short-delay filtering is carried out in cascade with long-delay filtering controlled by predetermined pitch and pitch predictor parameters.

3. An adaptive filtering method as defined in claim 1 or 2 including first order filtering with a transfer function $1-\mu z^{-1}$, $\mu<1$ in cascade with said short-delay filtering.

4. A method as defined in claim 1 or 2 wherein said parameters are predetermined at a transmitter by analysis of digital speech or audio signals before processing for transmittal to said receiver and said parameters are transmitted as side information to said receiver along with said digitally processed speech or audio signals.

5. A method as defined in claim 1 or 2 wherein said parameters are predetermined at said receiver by performing analysis of digitally processed speech or audio signals received.

6. A method as defined in claim 2 wherein said LPC parameters are predetermined at said transmitter by analysis of digital speech or audio signals before processing and transmitting as side information to said receiver, and said pitch and pitch predictor parameters are predetermined at said receiver by performing analysis of digitally processed speech or audio signals received.

7. A method as defined in any of claims 1 to 6 including automatic gain control of said digitally processed signal after filtering by computing a value $\sigma_2(n)$ proportional to volume of filtered speech or audio signals and a value $\sigma_1(n)$ proportional to volume of speech or audio signals before filtering and controlling the gain of the filtered speech or audio signals by a ratio of $\sigma_1(n)$ to $\sigma_2(n)$.

8. A method as claimed in claim 2 wherein postfiltering is accomplished by using a transfer function for said long-delay postfilter of the form

$$C_g \cdot \frac{1 + \gamma z^{-p}}{1 - \lambda z^{-p}}$$

where C_g is an adaptive scaling factor, and the factors γ and λ are determined according to the following formulas

$$\gamma = C_Z f(x),\ \lambda = C_P f(x),\ 0 < C_Z, C_P < 1$$

where

$$1 \text{ if } z > 1$$

$$f(z) = x \text{ if } U_{th} \leq x \leq 1$$

$$0 \text{ if } x < U_{th}$$

U_{th} is a threshold value and x can be either b_2 or $b_1+b_2+b_3$ depending upon whether a one-tap or three-tap pitch predictor is used.

Revendications

1. Procédé de filtrage adaptatif pour améliorer des signaux vocaux ou audio traités numériquement, dans un récepteur, par filtrage desdits signaux traités numériquement avec un filtrage produisant un bref retard, ledit filtrage introduisant un bref retard étant commandé par des paramètres de coefficients déterminés de prédiction linéaire (LPC); caractérisé en ce que ledit filtrage produisant un bref retard utilise une fonction de transfert à pôles zéros constituée par le rapport des deux fonctions de transfert tous pôles, les zéros de ladite fonction de transfert à pôles zéros ayant des rayons plus petits que les pôles correspondants.

2. Procédé de filtrage adaptatif selon la revendication 1, selon lequel ledit filtrage introduisant un bref retard est exécuté en cascade avec un filtrage produisant un long retard, commandé par des paramètres de pas et de prédicteurs de pas prédéterminés.

3. Procédé de filtrage adaptatif selon la revendication 1 ou 2, comprenant un filtrage du premier ordre avec une fonction de transfert $1-\mu z^{-1}$, $\mu < 1$ en cascade avec ledit filtrage produisant un bref retard.

4. Procédé de filtrage adaptatif selon la revendication 1 ou 2, selon lequel lesdits paramètres sont prédéterminés dans un émetteur - récepteur par analyse des signaux vocaux ou audio numériques avant le traitement pour l'émission vers ledit récepteur, et lesdits paramètres sont émis en tant qu'information secondaire vers ledit récepteur conjointement avec lesdits signaux vocaux ou audio traités numériquement.

5. Procédé de filtrage adaptatif selon la revendication 1 ou 2, selon lequel lesdits paramètres sont prédéterminés dans ledit récepteur par exécution d'une analyse de signaux vocaux ou audio traités numériquement, reçus.

6. Procédé de filtrage adaptatif selon la revendication 2, selon lequel lesdits paramètres LPC sont prédéterminés dans ledit émetteur par analyse de signaux vocaux ou audio numériques avant traitement et émission en tant qu'information secondaire audit récepteur, et lesdits paramètres de pas et de prédicteurs de pas sont prédéterminés dans ledit récepteur par exécution de l'analyse de signaux vocaux ou audio traités numériquement, reçus.

7. Procédé de filtrage adaptatif selon l'une quelconque des revendications 1 à 6, incluant une commande automatique de gain dudit signal traité numériquement après filtrage, par calcul d'une valeur $\sigma_2(n)$ proportionnelle au volume de signaux vocaux ou audio filtrés et d'une valeur $\sigma_1(n)$ proportionnelle au volume de signaux vocaux ou audio avant le filtrage, et commande du gain des signaux vocaux ou audio filtrés au moyen d'un rapport de $\sigma_1(n)$ à $\sigma_2(n)$.

8. Procédé selon la revendication 2, selon lequel un post-filtrage est exécuté en utilisant une fonction de transfert pour ledit filtre aval produisant un long retard, sous la forme

$$C_g \cdot \frac{1 + \gamma z^{-p}}{1 - \lambda z^{-p}}$$

C_g étant un facteur d'échelle adaptatif, et les facteurs γ et λ étant déterminés conformément aux formules suivantes

$$\gamma = C_z f(x),\ \lambda = C_p f(x),\ 0 < C_z, C_p < 1$$

avec

$$1 \text{ si } z > 1$$

$$f(z) = x \text{ si } U_{th} \leq x \leq 1$$

$$0 \text{ si } x < U_{th}$$

U_{th} étant une valeur de seuil et x pouvant être soit b_2, soit $b_1+b_2+b_3$ en fonction du fait qu'on utilise un prédicteur de pas à une prise ou à trois prises.

Patentansprüche

1. Adaptives Filterverfahren zum Verbessern digital verarbeiteter Sprach- oder Tonsignale bei einem Empfänger durch Filterung der digital verarbeiteten Signale mittels Kurz-Verzögerung-Filterung, wobei die Kurz-Verzögerung-Filterung durch vorbestimmte Linear-Prädiktionskoeffizientenparameter (LPC) gesteuert wird, dadurch gekennzeichnet, daß die Kurz-Verzögerung-Filterung eine Übertragungsfunktion mit Polen und Nullstellen verwendet, die aus dem Verhältnis von zwei Übertragungsfunktionen mit nur Polen besteht, wobei die Nullstellen der Übertragungsfunktion mit Polen und Nullstellen kleinere Radien aufweisen als entsprechende Pole.

2. Adaptives Filterverfahren nach Anspruch 1, worin die Kurz-Verzögerung-Filterung in Reihe mit Lang-Verzögerung-Filterung durchgeführt wird, die durch vorbestimmte Tonhöhen- und Tonhöhenprädiktionsparameter gesteuert wird.

3. Adaptives Filterverfahren nach Anspruch 1 oder 2, umfassend Filterung erster Ordnung mit einer Übertragungsfunktion $1-\mu z^{-1}$, $\mu<1$ in Reihe mit der Kurz-Verzögerung-Filterung.

4. Verfahren nach Anspruch 1 oder 2, worin die Parameter bei einem Sender vorbestimmt werden durch Analyse digitaler Sprach- oder Tonsignale vor Verarbeitung zur Übertragung an den Empfänger und daß die Parameter als Nebeninformation zusammen mit den digital verarbeiteten Sprach- oder Tonsignalen an den Empfänger übertragen werden.

5. Verfahren nach Anspruch 1 oder 2, worin die Parameter bei dem Empfänger vorbestimmt werden durch Ausführen einer Analyse empfangener digital verarbeiteter Sprach- oder Tonsignale.

6. Verfahren nach Anspruch 2, worin die LPC-Parameter bei dem Sender vorbestimmt werden durch Analyse digitaler Sprach- oder Tonsignale vor Verarbeitung und Übertragung als Nebeninformation an den Empfänger und die Tonhöhen- und Tonhöhenprädiktionsparameter bei dem Empfänger vorbestimmt werden durch Ausführen einer Analyse digital verarbeiteter empfangener Sprach- oder Tonsignale.

7. Verfahren nach einem der Ansprüche 1 bis 6, umfassend automatische Gewinnsteuerung des digital verarbeiteten Signals nach Filterung durch Berechnen eines Wertes $\sigma_2(n)$, der proportional ist zum Pegel von gefilterten Sprach- oder Tonsignalen, und eines Werts $\sigma_1(n)$, der proportional ist zum Pegel von Sprach- oder Tonsignalen vor Filterung, und Steuern des Gewinns der gefilterten Sprach- oder Tonsignale mit einem Verhältnis aus $\sigma_1(n)$ zu $\sigma_2(n)$.

8. Verfahren nach Anspruch 2, worin Nachfilterung erzielt wird durch Verwendung einer Übertragungsfunktion für die Lang-Verzögerung-Nachfilterung, welche die Gestalt

$$C_g \frac{1 + \gamma z^{-p}}{1 - \lambda z^{-p}}$$

aufweist, wobei C_g ein adaptiver Skalierungsfaktor ist und die Faktoren γ und λ gemäß der folgenden Formeln

$$\gamma = c_z f(x),\ \lambda = c_p f(x),\ 0 < c_z, c_p < 1$$

bestimmt werden, wobei

$$f(z) = \begin{cases} 1 & \text{wenn } z > 1 \\ x & \text{wenn } U_{th} \leq x < 1 \\ 0 & \text{wenn } x < U_{th} \end{cases}$$

ist und U_{th} ein Schwellenwert ist und x entweder b_2 oder $b_1 + b_2 + b_3$ sein kann, je nachdem ob ein Tonhöhenprädiktor mit einem Knotenanschluß oder mit drei Knotenanschlüssen verwendet wird.

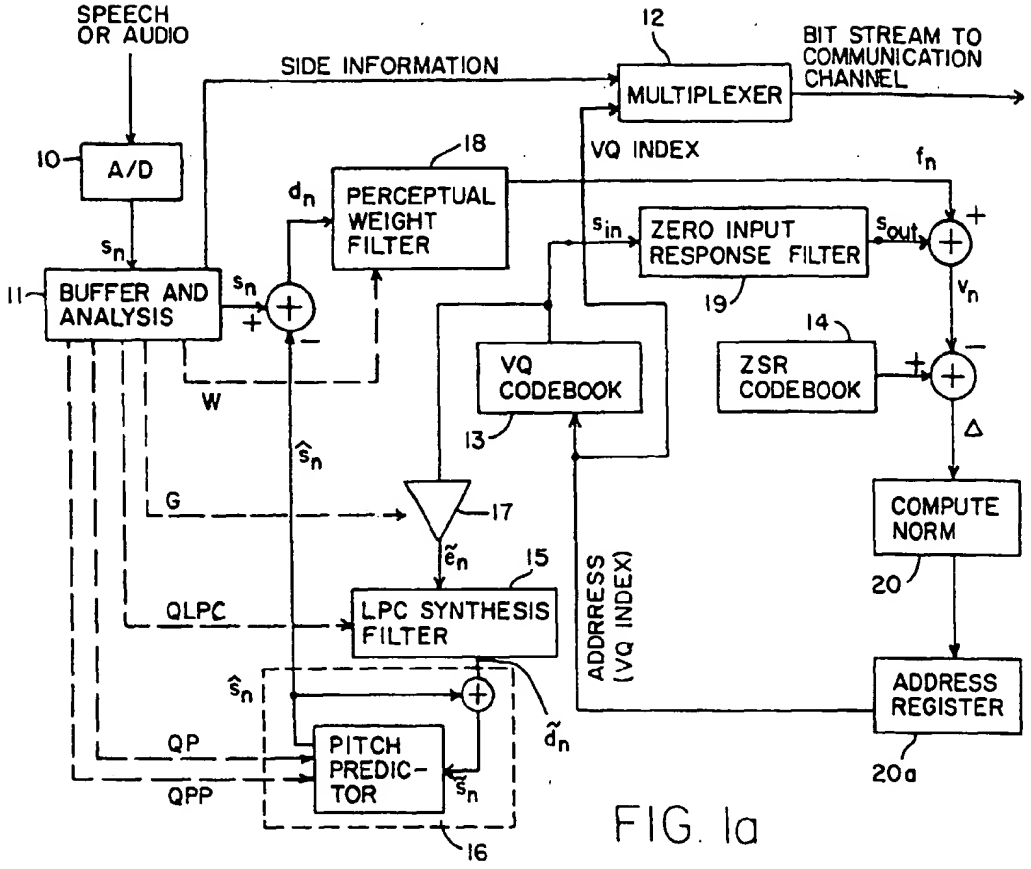
SPEECH OR AUDIO
SIDE INFORMATION
12
MULTIPLEXER
BIT STREAM TO COMMUNICATION CHANNEL
10
A/D
18
PERCEPTUAL WEIGHT FILTER
VQ INDEX
f_n
d_n
s_{in}
ZERO INPUT RESPONSE FILTER
s_{out}
s_n
11
BUFFER AND ANALYSIS
s_n
v_n
19
14
VQ CODEBOOK
ZSR CODEBOOK
W
13
$\hat{s}_n$
Δ
G
17
COMPUTE NORM
$\tilde{e}_n$
15
20
QLPC
LPC SYNTHESIS FILTER
ADDRESS (VQ INDEX)
$\hat{s}_n$
ADDRESS REGISTER
$\tilde{d}_n$
QP
PITCH PREDIC-TOR
20a
$\tilde{s}_n$
QPP
16

FIG. 1a

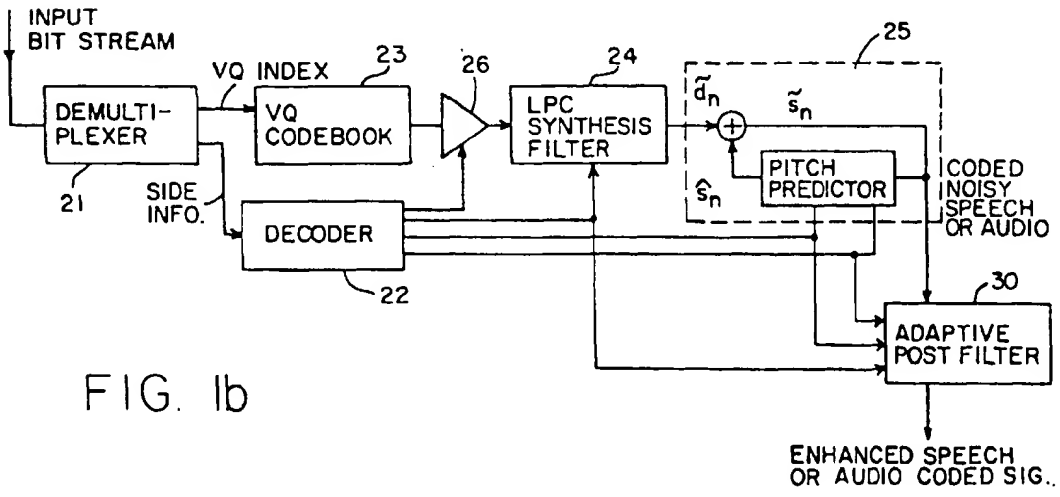
INPUT BIT STREAM
VQ INDEX
23
26
24
25
$\tilde{d}_n$
$\tilde{s}_n$
DEMULTI-PLEXER
VQ CODEBOOK
LPC SYNTHESIS FILTER
PITCH PREDICTOR
$\hat{s}_n$
21
SIDE INFO.
CODED NOISY SPEECH OR AUDIO
DECODER
22
30
ADAPTIVE POST FILTER
ENHANCED SPEECH OR AUDIO CODED SIG.

FIG. 1b

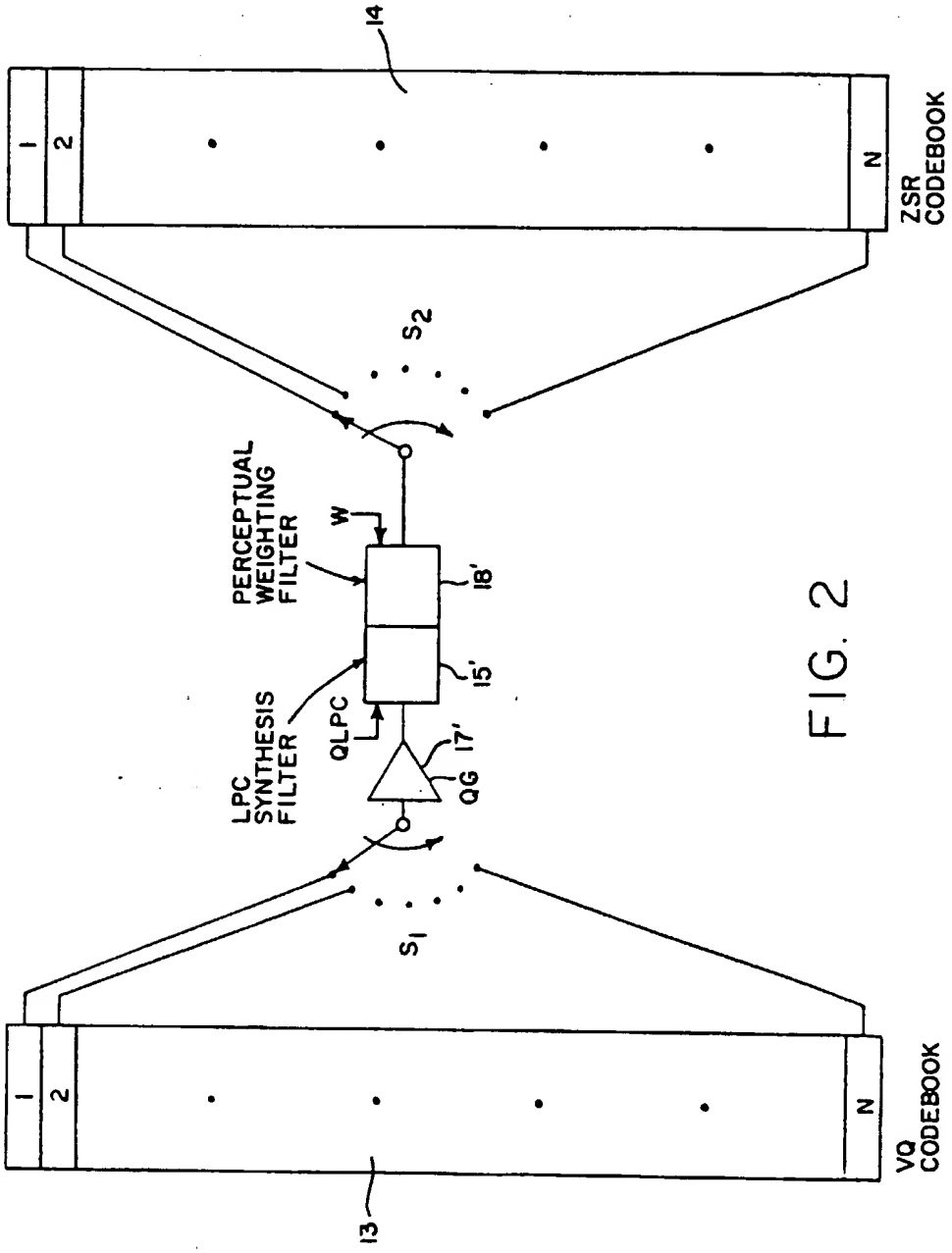
14
1
2
N
ZSR CODEBOOK
S_2
PERCEPTUAL WEIGHTING FILTER
W
18'
15'
LPC SYNTHESIS FILTER
QLPC
17'
QG
S_1
1
2
N
VQ CODEBOOK
13

FIG. 2

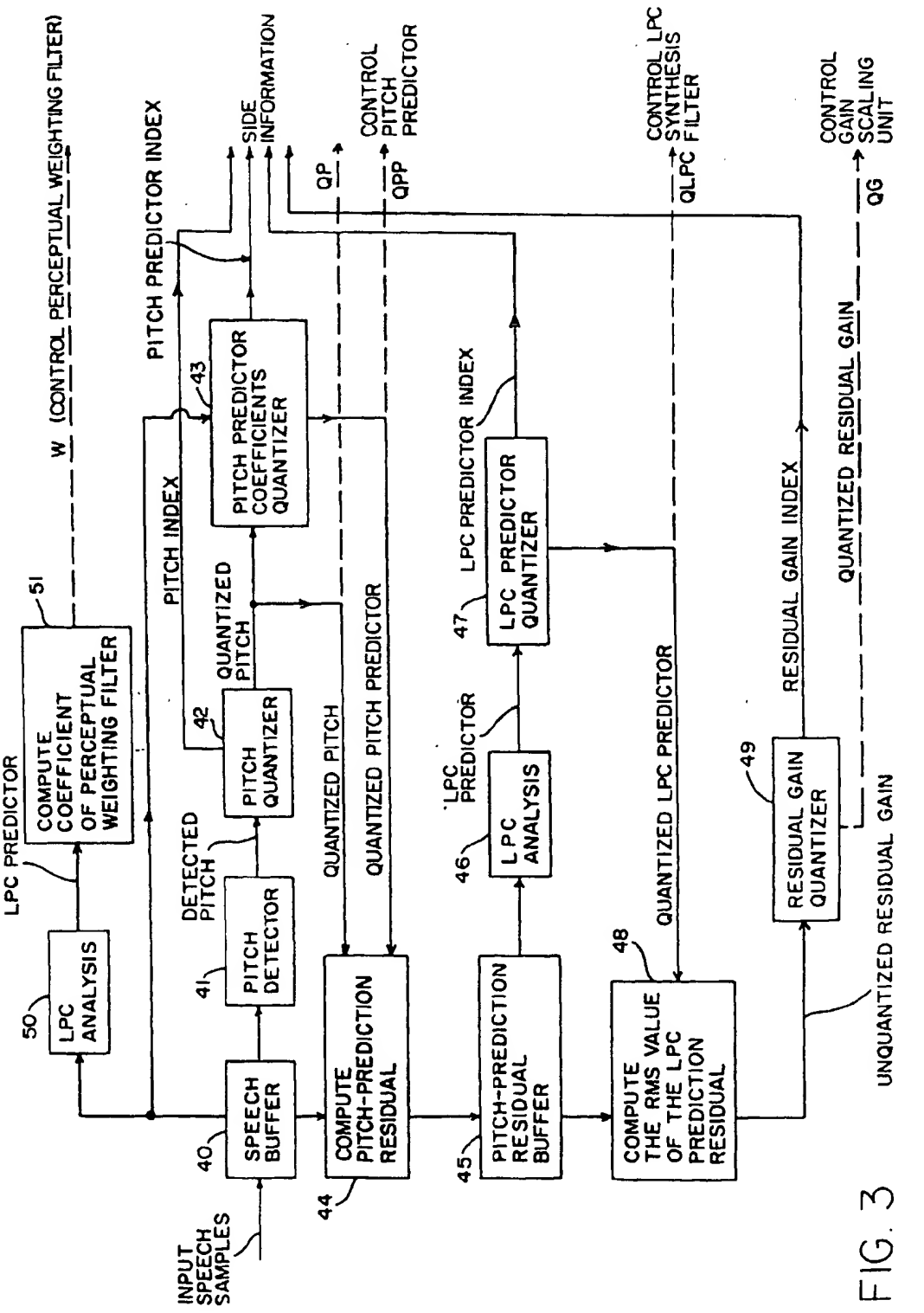
INPUT SPEECH SAMPLES
SPEECH BUFFER
40
LPC ANALYSIS
50
LPC PREDICTOR
COMPUTE COEFFICIENT OF PERCEPTUAL WEIGHTING FILTER
51
W (CONTROL PERCEPTUAL WEIGHTING FILTER)
PITCH DETECTOR
41
DETECTED PITCH
PITCH QUANTIZER
42
QUANTIZED PITCH
PITCH INDEX
PITCH PREDICTOR COEFFICIENTS QUANTIZER
43
PITCH PREDICTOR INDEX
SIDE INFORMATION
QP
CONTROL PITCH PREDICTOR
QPP
COMPUTE PITCH-PREDICTION RESIDUAL
44
QUANTIZED PITCH
QUANTIZED PITCH PREDICTOR
PITCH-PREDICTION RESIDUAL BUFFER
45
LPC ANALYSIS
46
LPC PREDICTOR
LPC PREDICTOR QUANTIZER
47
LPC PREDICTOR INDEX
CONTROL LPC SYNTHESIS FILTER
QLPC
COMPUTE THE RMS VALUE OF THE LPC PREDICTION RESIDUAL
48
QUANTIZED LPC PREDICTOR
RESIDUAL GAIN QUANTIZER
49
RESIDUAL GAIN INDEX
QUANTIZED RESIDUAL GAIN
CONTROL GAIN SCALING UNIT
QG
UNQUANTIZED RESIDUAL GAIN

FIG. 3

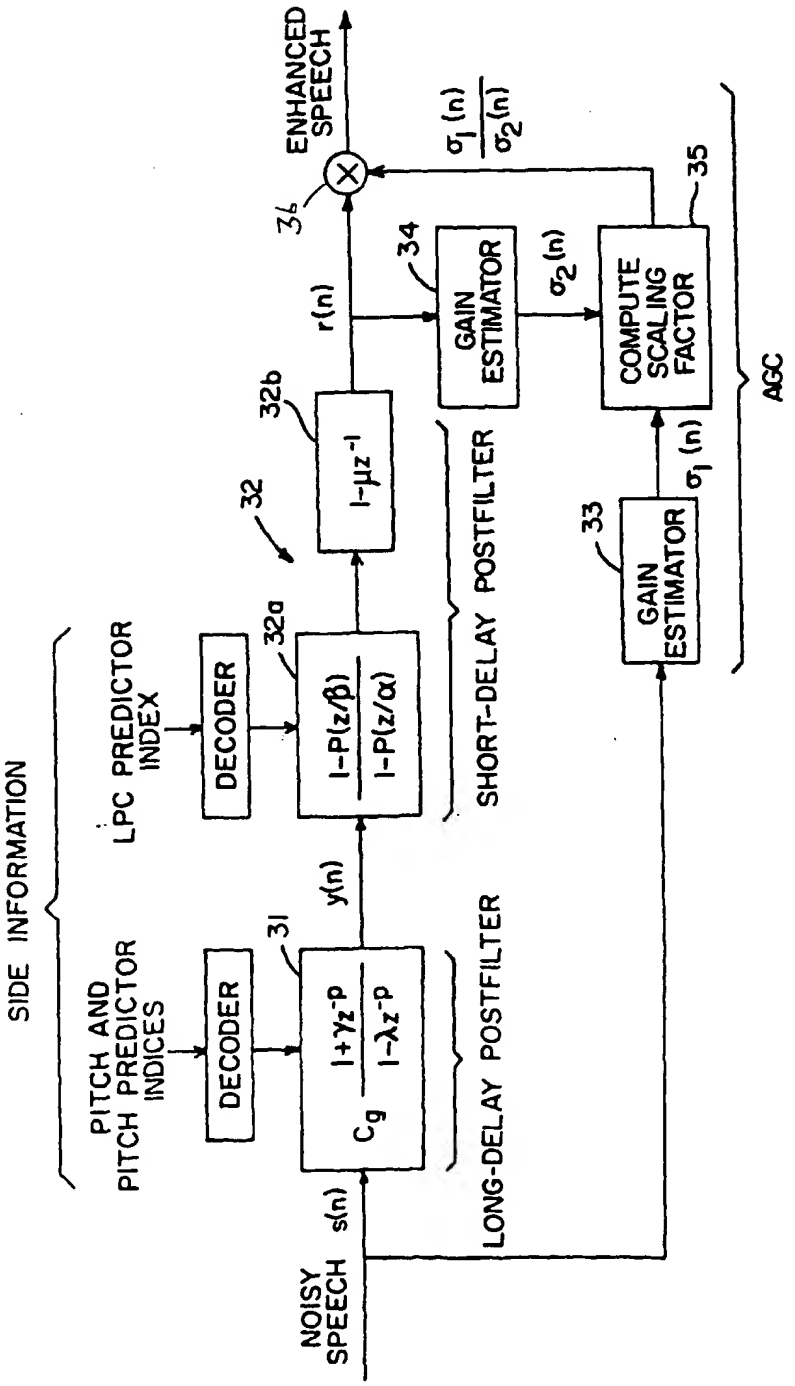
SIDE INFORMATION
PITCH AND PITCH PREDICTOR INDICES
LPC PREDICTOR INDEX
DECODER
DECODER
31
32
32a
32b
$C_g \frac{1+\gamma z^{-p}}{1-\lambda z^{-p}}$
$\frac{1-P(z/\beta)}{1-P(z/\alpha)}$
$1-\mu z^{-1}$
y(n)
r(n)
s(n)
NOISY SPEECH
LONG-DELAY POSTFILTER
SHORT-DELAY POSTFILTER
34
GAIN ESTIMATOR
$\sigma_2(n)$
COMPUTE SCALING FACTOR
35
AGC
33
GAIN ESTIMATOR
$\sigma_1(n)$
36
ENHANCED SPEECH
$\frac{\sigma_1(n)}{\sigma_2(n)}$

FIG. 4

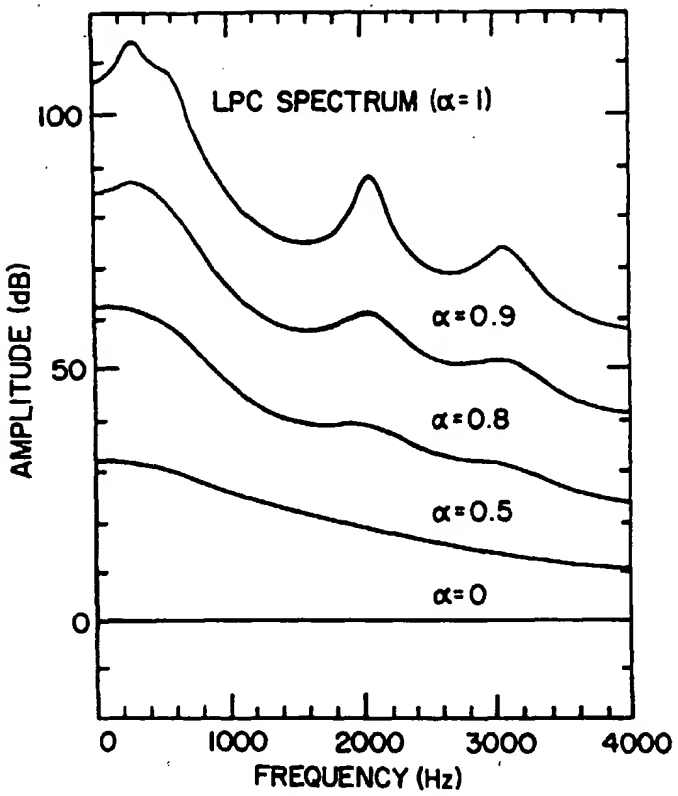
LPC SPECTRUM (α=1)
α=0.9
α=0.8
α=0.5
α=0
AMPLITUDE (dB)
100
50
0
0
1000
2000
3000
4000
FREQUENCY (Hz)

FIG. 5

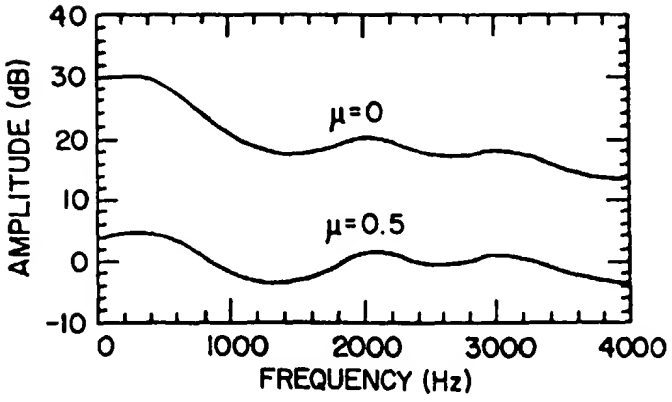
μ=0
μ=0.5
AMPLITUDE (dB)
40
30
20
10
0
-10
0
1000
2000
3000
4000
FREQUENCY (Hz)

FIG. 6